



DTO Java Record #1

Transcripción

[00:00] Bien. Ya tenemos nuestros parámetros ya recibidos aquí en nuestro back end, nuestro API. Y la pregunta hora es cómo podemos acceder directamente a un campo específico dentro del body, dentro del payload que estamos recibiendo.

[00:18] Por ejemplo, a mí se me ocurre que si lo que yo hago es ponerle el nombre del campo que yo quiero recibir aquí, por ejemplo, si yo quiero recibir el nombre puede ser que si yo le pongo nombre automáticamente Spring, me lo van a mapear y me va a imprimir aquí solo el nombre Entonces vamos a guardar para ver qué es lo que sale.

[00:41] Esperamos un poco a que recargue nuestro servidor y lanzamos el request nuevamente aquí, lanzamos nuevamente 200 éxito pero insistimos en el mismo problema. Imprime el JSON completo. ¿Esto es por qué? Porque Spring no es adivino como para saber que queremos acceder a este parámetro.

[01:04] Lo único que hicimos fue básicamente cambiar el nombre del parámetro que estábamos recibiendo como body. Entonces vamos a regresar a parámetro y vamos a explorar otras formas de obtener esto, pero ya a nivel de objeto, cómo podemos tratar como objeto, estamos con Java. Tenemos que pensar siempre en objetos.

[01:25] Para eso es lo que yo voy a hacer es crear aquí un récord. Si no están familiarizados con record, son básicamente un tipo de objeto de Java, inmutable, que te permite crear dinámicamente o rápidamente campos

solamente para llenarlos y en tiempo de compilación y lo que va a hacer Java es generarte una clase propiamente dicha con el constructor, getter, setters y lo que fuera necesario.

[01:55] Para esto entonces lo que vamos a hacer aquí es crear un record llamado DatosMedico. DatosRegistroMedico, porque estos son los otros datos que usamos para el registro de médico y seleccionamos record. Entonces en este record lo que va entre los paréntesis son los parámetros que necesitamos, por ejemplo yo sé que necesito nombre que es tipo string, y perfecto, el DatosRegistroMedico va a recibir este parámetro aquí.

[02:21] Vamos a nuestro payload para ver qué necesitamos. Necesitamos nombre, email, documento. Entramos aquí a nombre string email, string documento. Y tenemos especialidad. Pero especialidad si vemos el modelo que recibimos del departamento de UX, vemos que es un dropdown.

[02:45] En nuestra clínica por regla de negocio vamos a tener cuatro especialidades, que son pediatría, ortopedia, cardiología y ginecología. Entonces esto es un claro caso de 1 en 1. Vamos a usar 1 en 1 para esto. Entonces le vamos a dar aquí especialidad y vemos que nos va a fallar porque no tenemos pues ningún objeto llamado especialidad.

[03:12] Entonces le vamos a dar crear especialidad, pero lo que vamos a hacer ahora es poner todo esto dentro de un paquete, que sea el paquete médico. ¿Por qué? Porque estamos hablando ahora propiamente de datos de registro de un médico y para tener cierto orden en nuestro proyecto vamos a crearlo.

[03:32] Entonces le damos okay, tenemos nuestro paquete médico, datos registro médico ya lo movió y vamos a crear aquí un ENAM. New Java class, especialidad tipo ENAM y listo. ¿Qué vamos a hacer ahora? Vamos a escribir las especialidades que tengo que son ortopedia, cardiología, ginecología y pediatría. Vamos a revisar que esté bien escrito.

[04:08] Entonces nuestro ENAM ya está creado, venimos aquí, importamos nuestro ENAM que es de aquí y ya tenemos esto listo. ¿Qué más necesitamos para registrar nuestro médico? La dirección Pero dirección vemos que lleva como un objeto y yo no lo voy a crear dentro del paquete médico porque yo asumo que los pacientes también necesitan ir a una dirección, por regla del negocio.

[04:37] Entonces lo que voy a hacer aquí es exactamente igual, voy a crear otro récord llamado DatosDirección. Entonces le doy aquí, creamos, no lo voy a agregar aun. ¿Y en la dirección qué tenemos? Tenemos calle, distrito, entonces vamos a tener que agregar un String calle, String distrito, String ciudad. Vamos a ponerle un entero al número y un complemento.

[05:26] Por el momento voy a manejar todo con string, solo para tenerlo mucho más simplificado y no preocuparme por el tipo de dato que estamos recibiendo, validación ya lo vamos a ver después. Entonces ya tengo mis datos de dirección aquí: calle, distrito, ciudad, número y complemento.

[05:41] Y al igual que cualquier otro objeto, aquí lo que voy a ponerle es un DatosDirección. Entonces con esto ya tengo todo mi récord creado. Recuerden, este record está disponible en las últimas versiones de Java, lo que hace básicamente es crear una clase normal como la que conocen, pero ya cuando compilamos el código.

[06:06] Eso es solo para simplificar y no tener que escribir nuevamente el mismo código. Ahora regresamos al control y vamos a cambiar el tipo de dato que estamos esperando aquí. Y le decimos ahora: “quiero un DatosRegistroMedico como parámetro”. Y le vamos a cambiar el nombre a datosRegistroMedico para que sea mucho más entendible. Recuerden que su código tiene que ser siempre legible.

[06:31] Voy a borrar también esto de aquí, porque ya no lo necesitamos y vamos a ver ahora qué es lo que sale. Esperamos a que cargue nuestro servidor

nuevamente. Vemos que ya cargó y vamos a limpiar esto. Vamos a ver ahora qué es lo que sale aquí.

[06:49] Le damos a enviar y recibimos un bad request aquí. Entonces esto es bueno. ¿Por qué? Porque está llegando el request aquí, vemos que aquí Spring consigue resolver, pero nos está diciendo aquí que no puede deserializar el valor del tipo especialidad de ortopedia. Y es el mismo error, que nos está dando aquí en la consola, porque estamos devolviendo el error tal cual el servidor nos lo está dando.

[07:17] No estamos haciendo ningún tratamiento en la respuesta todavía. ¿Cuál es el problema aquí? Bueno, tengo especialidad ortopedia, pero en nuestro enum, que lo tenemos aquí, está en mayúsculas, porque los enum por buena práctica siempre van en mayúscula.

[07:34] Entonces aquí tienen dos opciones, la primera puede ser coordinar con el equipo que está desarrollando el front end, para que les envíe este tipo de datos siempre en mayúscula, y la otra que no la vamos a aplicar, pero para que la sepan, puede ser ponerle el nombre aquí.

[07:52] Crear un constructor para el enum, de modo que automáticamente puede hacer el mapeo entre el parámetro y el valor. No lo vamos a hacer por ahora porque sería complicarnos demasiado. Lo que vamos a hacer es el camino más fácil, que es cambiar esto por mayúsculas.

[08:07] Nuevamente ahora ortopedia está en mayúscula. Vamos a enviar Y tenemos nuevamente otro bad request. ¿Y qué es lo que tenemos aquí? ¿Qué error tenemos ahora? El error es que no hemos guardado aun. No olviden guardar para que automáticamente web tools se dispare y nos reinicie nuestro servidor nuevamente.

[08:32] Entonces enviamos nuevamente, tenemos el bad request y vemos qué es lo que nos está diciendo ahora: invalid utf-8 start byte.

